上級編

第47講 パリティチェック

例題

以下の条件で，偶数パリティ方式を用いた，誤り検出を行うプログラムを作成する。

・偶数パリティとは，誤り検出方式の一つで，ビット列中に含まれる「1」の個数が奇数個ならパリティビットを「1」に，偶数個なら「0」にする方式である。

・例えば，「1010」というビット列は 1 の数が偶数個なので，末尾にパリティビットを加えると「10100」となる。

・一方，「1000」というビット列は 1 の数が奇数個なので，末尾にパリティビットを加えると「10001」となる。

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
count = 0
bit_retsu = input('パリティビットを含めたビット列を入力してください：')

for i in range(len(bit_retsu) - 1):
  if bit_retsu[i] == '1':
    count = count + 1

if ________:
  parity_bit = '0'
else:
  parity_bit = '1'
# パリティビットの検証
if parity_bit == bit_retsu[-1]: # bit_retsu[-1] は，bit_retsu 配列の最後の要素を取得
  print('ビット列は正しいです。パリティビット:', parity_bit)
else:
  print('ビット列は誤っています。正しいパリティビットは ', parity_bit , 'です。')
```

① count % 2 == 0

② count % 2 == 1

③ count == 0

④ count / 2 == 0

＜MEMO＞

上級編　第 47 講　確認問題

以下の条件で，JAN コードのチェックディジットを求め，誤りチェックを行うプログラムを作成する。

・JAN コードとは，バーコードの規格の一つであり，どのメーカーのどの商品であるかを示す国際標準の商品識別コードである。

・13 桁の JAN コードの場合，末尾 1 桁がチェックディジットである。

1 234567 890128

チェックディジット

・チェックディジットとは，入力の誤りを検出したり，捏造を防止したりするために，一定のルールに従って元の符号に加えられる数値や記号のことである。

・JAN コードのチェックディジットは，次のような手順で算出する。

(1)　桁数が奇数桁のコードの和を求める（チェックディジットを除く）。

(2)　桁数が偶数桁のコードの和を求め，それを 3 倍する。

(3)　(1)の結果と(2)の結果の和を求める。

(4)　(3)の結果の 1 の位の数を 10 から引いた結果の 1 桁目がチェックディジットである。　例：1 の位が 3 のとき　10-3=7

・入力が 13 桁でなかった場合は exit() という命令を用いて，その時点でプログラムを終了する。

以下のプログラムの空欄に入る最も適切なものを，あとの①～④のうちから一つ選びなさい。

```
Jan_code = input('13 桁の JAN コードを入力してください：')

if len(Jan_code) != 13:
  print('入力されたコードは 13 桁ではありません。')
  exit()

odd_sum = 0 # 奇数桁の合計値を格納
even_sum = 0 # 偶数桁の合計値を格納

for i in range(12):
  if i % 2 == 0:
    odd_sum = odd_sum + int(Jan_code[i])
  else:
    even_sum = even_sum + int(Jan_code[i])

total = [          ]

ones_place = total % 10

if ones_place == 0:
  check_digit = 0
else:
  check_digit = 10 - ones_place

# チェックディジットの検証
if check_digit == int(Jan_code[-1]):
  print('JAN コードは正しいです。チェックディジット：', check_digit)
else:
  print('JAN コードは誤っています。正しいチェックディジットは', check_digit, 'です。')
```

① odd_sum * 3 + even_sum

② odd_sum + even_sum * 3

③ odd_sum * even_sum

④ odd_sum + 3 + even_sum

上級編　例題　解答

第34講　コインの枚数

例題　解答：④

第35講　駐車料金の判定

例題　解答：④

第36講　割引のある料金

例題　解答：④

第37講　テストの集計

例題　解答：①

第38講　じゃんけん

例題　解答：①

第39講　サイコロの出目

例題　解答：③

第40講　中央値の判定

例題　解答：①

第41講　素数判定

例題　解答：③

第42講　素因数分解

例題　解答：②

第43講　倍数の判定・FizzBuzz

例題　解答：④

第44講　バブルソート・選択ソート

例題　解答：③

第45講　グラフ理論

例題　解答：②

第46講　待ち行列

例題　解答：②

第47講　パリティチェック

例題　解答：①

第48講　すごろくゲーム

例題　解答：④

上級編　第 47 講　確認問題　解答

正解：②

解説：

問題文のチェックディジットの求め方の手順⑴⑵から，左から数えたとき，奇数番目になる桁のときと，偶数番目になる桁のときで処理を分ける必要があることが分かる。

11 行目の if 文では，`if i % 2 == 0` とあるので，これは Jan_code の添字 i が偶数（0 番目，2 番目…），つまり，桁数が奇数桁（1 桁目, 3 桁目…）の条件を指定している。この条件が True であるとき，奇数の桁数の和を表す変数 odd_sum に奇数桁の値を加算している。

13 行目 else 文では，桁数が偶数桁の場合となるので，偶数桁の値を加算して，偶数の桁数の和を表す変数 even_sum に代入している。ここまでの処理で，奇数桁の合計値 odd_sum と偶数桁の合計値 even_sum が求められた。

続いて，問題文のチェックディジットの求め方の手順⑶では，「⑴の結果と⑵の結果の和を求める」とあるため，`odd_sum + even_sum` の式となる。ただし, 注意したいのは，手順⑵で偶数桁の合計値を 3 倍する，と指定があるためこの式を `odd_sum + even_sum *3` とする必要がある。よって，正解は ② `odd_sum + even_sum * 3` である。

```
Jan_code = input('13桁のJANコードを入力してください：')

if len(Jan_code) != 13:
  print('入力されたコードは13桁ではありません。')
  exit()

odd_sum = 0 # 奇数桁の合計値を格納
even_sum = 0 # 偶数桁の合計値を格納

for i in range(12):
  if i % 2 == 0:
    odd_sum = odd_sum + int(Jan_code[i])
  else:
    even_sum = even_sum + int(Jan_code[i])

total = odd_sum + even_sum * 3

ones_place = total % 10

if ones_place == 0:
  check_digit = 0
else:
  check_digit = 10 - ones_place

# チェックディジットの検証
if check_digit == int(Jan_code[-1]):
  print('JANコードは正しいです。チェックディジット：', check_digit)
else:
  print('JANコードは誤っています。正しいチェックディジットは', check_digit, 'です。')
```

実行結果

```
13桁のJANコードを入力してください：1234567890123
JANコードは誤っています。正しいチェックディジットは 8 です。
```